

Maximum Linear Forests in Trees^{*}

Jian Wang[†]

College of Mathematics

Taiyuan University of Technology, Taiyuan, 030024, P.R.China

Abstract. For a graph G , let $t(G)$ (resp. $f(G)$) be the maximum number of vertices in an induced subgraph of G that is a tree (resp. a forest). Erdős, Saks and Sós show that line graphs of specific trees almost have smallest values of $t(G)$ over all graphs thirty years ago. It is shown that if G is the line graph of tree T , then $t(G)$ equals to the diameter of T . An interesting question is that how large the gap can be between $t(G)$ and $f(G)$ over all line graphs of trees. In this paper, we give lower and upper bounds on $f(G)$ over line graphs of trees with given diameter and line graphs of full k -ary trees. We also construct extremal graphs that achieve these bounds. We find that finding maximum induced forests in line graphs is equivalent to finding maximum linear forests in corresponding original graphs. Although finding maximum linear forests in graphs is NP-hard in general, we proposed a polynomial time algorithm for finding maximum linear forests in trees.

Keywords: maximum induced forests, maximum linear forests, line graphs, trees

1 Introduction

For a graph G , let $t(G)$ be the maximum number of vertices in an induced subgraph of G that is a tree. The problem of bounding $t(G)$ in the connected graph G was first introduced thirty years ago by Erdős, Saks and Sós [1]. In their paper, Erdős, Saks and Sós studied the relationship between $t(G)$ and several natural parameters of the graph G . They were able to obtain asymptotically tight bounds on $t(G)$ when either the number of edges or the independence number of G were known. Their result show that $t(G)$ can be surprisingly small over graphs with n vertices and m edges. Given a graph G , its line graph $L(G)$ is a graph such that each vertex of $L(G)$ represents an edge of G and two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G . Erdős, Saks and Sós use line graphs of special trees to construct graphs for which $t(G)$ is small. Besides, Erdős, Saks and Sós also considered the problem of estimating the size of maximum induced tree in K_r -free graphs. Recently, Jacob Fox, Po-Shen Loh and Benny Sudakov improve the results on lower bounds of maximum induced trees in K_r -free graphs by a very beautiful proof [2].

One can also study induced forests rather than trees in graphs. Let $f(G)$ be the size

^{*}The work was supported by NNSF of China (No.11671299, No.61502330).

[†] Corresponding author: wangjian01@tyut.edu.cn

of maximum induced forest in graph G . Trivially we have $f(G) \geq t(G)$, but it appears that the size of the maximum induced forest in a graph is more closely related to another parameter. The independent number $\alpha(G)$ of graph G is the maximum size of independent set in G . Since every independent set is also an induced forest and every induced forest is bipartite, $f(G)$ is at least $\alpha(G)$ and at least $2\alpha(G)$. It turns out that the largest induced forest can be much larger than the largest induced tree in some graphs. Since line graphs of special trees have small value of $t(G)$, it is interesting to consider upper bounds on $f(G)$ over all line graphs of trees with given value of $t(G)$. In this paper, we prove that finding maximum induced forests in line graphs is equivalent to finding maximum linear forests in original graphs. A *linear forest* in a graph G is a vertex disjoint union of simple paths of G . A *maximum linear forest* in G is a linear forest in G with maximum number of edges. By using this fact, we find lower and upper bounds on $f(G)$ over all line graphs of trees with given diameter. The extremal graphs that achieve these bounds are also constructed. It is shown that the extremal trees that achieve upper bounds are path-like, which means lots of vertices in them have degree two. Thus, we further consider bounds on $f(G)$ over all line graphs of full k -ary trees. Moreover, maximum induced forests in line graphs of perfect k -ary trees are determined.

A subset $F \subset V(G)$ is called a *feedback vertex set* if the subgraph $G - F$ is a forest. The minimum cardinality of a feedback vertex set (or decycling set) is called *feedback number* (or *decycling number*) of G , which is proposed first by Beineke and Vandell [4]. Finding the maximum induced forest of a graph G is equivalent to finding minimum feedback vertex set of G , since the sum of the two orders equals the order of G . Apart from its graph-theoretical interest, the minimum feedback vertex set problem has important application to several fields. For example, the problems are in operating systems to resource allocation mechanisms that prevent deadlocks, in artificial intelligence to the constraint satisfaction problem and Bayesian inference, in synchronous distributed systems to the study of monopolies and in optical networks to converters placement problem.

In fact, the problem of finding minimum feedback vertex set is NP-hard for general graphs [6] (also see [5]). The best known approximation algorithm for this problem has approximation ratio 2 [3]. There are also polynomial time algorithms for a number of special graph classes, such as reducible graphs [7], cocomparability graphs [11], permutation graphs [9], convex bipartite graphs [12], cyclically reducible graphs [8], interval graphs [10] and AT-free graphs [13] et al. In this paper, we show that finding minimum feedback vertex set is NP-hard for line graphs. However, there is a polynomial time algorithm for line graphs of trees. Since maximum induced forests in line graph $L(G)$ correspond to maximum linear forests in G . The problem to find maximum linear forests in graphs is equivalent to a specific Traveling Salesman Problem. By using this fact, we prove that finding maximum linear forests in graphs is NP-complete in general. For maximum linear forests in trees, we find a dynamic programming algorithm which runs in polynomial time.

The rest of this paper is organized as follows. In Section 2, we show that finding maximum induced forests in line graphs is equivalent to finding maximum linear forests in original graphs. In Section 3, we give lower and upper bounds on $f(G)$ over line graphs of trees with given diameter. In Section 4, we propose a recursive structure on maximum linear forests in rooted trees which implies a dynamic programming algorithm which runs in polynomial time. Finally, we determine the maximum linear forests in perfect k -ary trees in Section 5.

2 Maximum induced forests in line graphs

In this section, we first prove that the maximum induced forests in line graphs correspond to maximum linear forests in original graphs. Then we propose another graph parameter which is related to hamiltonian property of graphs.

Lemma 2.1. *A vertex-disjoint path P in G is longest if and only if $L(P)$ is a maximum induced tree in line graph $L(G)$. A linear forest F in G is maximum if and only if $L(F)$ is a maximum induced forest in line graph $L(G)$.*

Proof. Since line graphs are claw-free. It means that they have no induced $K_{1,3}$. So do their induced trees and induced forests. However, claw-free trees are paths and claw-free forests are unions of paths (linear forests). Thus, every induced trees of line graphs are induced paths and every induced forests of line graphs are induced linear forests.

Moreover, every line graph of a vertex-disjoint path in G is an induced path in $L(G)$ and every induced path in $L(G)$ is a line graph of a vertex-disjoint path in G . If $P = v_1e_1v_2e_2 \dots v_le_lv_{l+1}$ is a vertex-disjoint path in G , in which v_i 's are vertices and e_j 's are edges in G . Then, we shall show $L(P) = (e_1, e_2, \dots, e_l)$ is an induced path in $L(G)$. Otherwise, assume that (e_j, e_k) forms an edge in $L(G)$ and $k > j + 1$. Then e_j and e_k share a common ending point in G . We have $\{v_j, v_{j+1}\} \cap \{v_k, v_{k+1}\} \neq \emptyset$, which contradicts with path P is vertex-disjoint. If H is an induced path in line graph $L(G)$. Let e_1, e_2, \dots, e_l are l consecutive vertices in H . Clearly, $P = (e_1, e_2, \dots, e_l)$ is a path in G and $H = L(P)$. Let

$$\begin{cases} \mathcal{P} = \{\text{all paths in graph } G\}, \\ L(\mathcal{P}) = \{L(P) : P \in \mathcal{P}\}, \\ \mathcal{T} = \{\text{all induced trees in graph } L(G)\}; \end{cases}$$

and

$$\begin{cases} \mathcal{L} = \{\text{all linear forests in graph } G\}, \\ L(\mathcal{L}) = \{L(F) : F \in \mathcal{L}\}, \\ \mathcal{F} = \{\text{all induced forests in graph } L(G)\}. \end{cases}$$

It implies that $L(\mathcal{P}) = \mathcal{T}$ and $L(\mathcal{L}) = \mathcal{F}$. Thus, a vertex-disjoint path P in G is longest if and only if $L(P)$ is a maximum induced tree in line graph $L(G)$ and a linear forest F in G is maximum if and only if $L(F)$ is a maximum induced forest in line graph $L(G)$. ■

The number of edges in maximum linear forests of graph G is denoted by $l(G)$. Clearly, linear forests of G have at most $n - 1$ edges. Therefore, we have the following corollary.

Corollary 2.1. *For any graph G with n vertices, $f(L(G)) \leq n - 1$.*

A *hamiltonian cycle* is a cycle that visits each vertex exactly once (except for the vertex that is both the start and end, which is visited twice). A graph that contains a hamiltonian cycle is called a *hamiltonian graph*. For any graph G , the least number of edges that should be added to make G hamiltonian is denoted by $h(G)$. Obviously, if G is a hamiltonian graph, then $h(G) = 0$. We show that $h(G)$ and $l(G)$ have the following relation.

Lemma 2.2. *For any graph G with n vertices, if $h(G) > 0$, then $l(G) = n - h(G)$; if $h(G) = 0$, then $l(G) = n - 1$.*

Proof. If $h(G) = 0$, then G is hamiltonian. By Cor.2.1, we have $l(G) = n - 1$. If $h(G) > 0$, by adding $h(G)$ edges graph G becomes to a hamiltonian graph G' . Then G' has a hamiltonian cycle C . Remove all edges in $E(G') - E(G)$ from C , we get a linear forest of G . Thus, $l(G) \geq n - h(G)$. Conversely, if G has a linear forest F with $l(G)$ edges, then n vertices are connected to $n - l(G)$ components each of which is a path or a isolated vertex. Then arrange these $n - l(G)$ components on a cycle and add $n - l(G)$ edges, we get a cycle with n vertices. It means that $h(G) \geq n - l(G)$. Therefore, $l(G) = n - h(G)$. ■

For any graph G on n vertices, define a weighted complete graph K_n as follows. Let $w(e) = 0$ for $e \in E(G)$ and let $w(e) = 1$ for $e \notin E(G)$. Then $h(G)$ can be determined by solving a a specific Traveling Salesman Problem on this weighted complete graph. Moreover, to determine whether a graph has a hamiltonian path or not is equivalent to determine whether $l(G)$ equals to $n - 1$ or not. The fact implies that finding maximum linear forests in graphs is NP-hard.

3 Line graphs of trees

Let T be a tree with n vertices. An *inner vertex* is a vertex of degree at least two. Similarly, an *outer vertex* (or a *leaf*) is a vertex of degree one. Then the vertices of T can be partitioned into the set of leaves V_{out} and the set of inner vertices V_{in} . The cardinality of V_{out} is denoted by $out(T)$. For any inner vertex v , let $ex(v)$ be zero if v has at most two neighbors of degree less than three; let $ex(v)$ be $k - 2$ if v has k neighbors of degree less than three. Then we have the following lemma.

Lemma 3.1. *For any tree T with n vertices,*

$$\left\lceil \frac{out(T) + \sum_{v \in V_{in}} ex(v)}{2} \right\rceil \leq h(T) \leq out(T) - 1.$$

Proof. Since T is a tree. By adding $h(T)$ edges on T , we get a hamiltonian graph G . Since G has a hamiltonian cycle C , each leaf of T is incident to a new edge in C . For any inner vertex v , if $ex(v)$ is greater than zero, then at most two adjacent edges incident to v are in C . It means that at least $ex(v)$ degree ≤ 2 neighbors of v have at most one adjacent edge of T in C . Each of these neighbors is incident to a new edge in C . Thus, at least $out(T) + \sum_{v \in V_{in}} ex(v)$ vertices are incident to new edges. Therefore, we have

$$h(T) \geq \left\lceil \frac{out(T) + \sum_{v \in V_{in}} ex(v)}{2} \right\rceil.$$

On the other hand, we can get a hamiltonian cycle by adding edges in T according to the following procedure. Firstly, we choose two leaves u, v of T and add an edge between them. Let G_0 be the graph $T + uv$. Then G_0 contains a cycle C_0 formed by new edge uv and the unique path $P_{u,v}$ in T . Let T_0 be the graph obtained from G_0 by contracting cycle C_0 , or

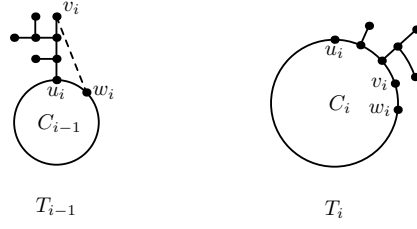


Figure 1: An example from tree T_{i-1} to T_i .

$T_0 = G_0 \cdot C_0$. We denote the contracted vertex by C_0 in T_0 . Since T_0 can also be obtained from T by contracting path P_{uv} and by contracting edges in a tree we cannot get a cycle. Therefore, T_0 is a tree with $\text{out}(T) - 2$ leaves besides vertex C_0 . Now choose a leaf v_1 from T_0 . Let $P_{C_0v_1}$ be the unique path between C_0 and v_1 . Suppose $P_{C_0v_1}$ in T_0 is $P_{u_1v_1}$ in T in which u_1 is a vertex in C_0 and u_1w_1 is an edge in C_0 . Then by adding edges v_1w_1 , we get a larger cycle $C_1 = C_0 - u_1w_1 + P_{u_1v_1} + v_1w_1$ in T . Now let T_1 be the graph obtained from T by contracting C_1 . Then T_1 can also be obtained from T_0 by contracting $P_{C_0v_1}$ and T_1 is a tree with $\text{out}(T) - 3$ leaves besides vertex C_1 . Now choose a leaf v_2 from T_1 . Let $P_{C_1v_2}$ be the unique path between C_1 and v_2 . $P_{C_1v_2}$ in T_1 is $P_{u_2v_2}$ in T in which u_2 is a vertex in C_1 and u_2w_2 is an edge in C_1 . Then by adding edges v_2w_2 , we get a larger cycle $C_2 = C_1 - u_2w_2 + P_{u_2v_2} + v_2w_2$ in T . Do this procedure repeatedly, an example from tree T_{i-1} to tree T_i has shown in Fig.1. Since each step we get a tree with leaves less than 1. The procedure has to be stopped when the contracted tree has only one vertex. Then we get a hamiltonian cycle by adding $\text{out}(T) - 1$ edges in T . Thus, $h(T) \leq \text{out}(T) - 1$. ■

Now we introduce a operation on leaves of trees that does not decrease $l(T)$. For any two leaves u_i, u_j of T , suppose their neighbors are w_i, w_j . We define Leaf-Exchange operation on T as removing edge w_iu_i from T and adding edge u_ju_i , the obtained tree is denoted by $T[u_i \rightarrow u_j]$.

Lemma 3.2. For any two leaves u_i, u_j of T , $l(T[u_i \rightarrow u_j]) \geq l(T)$.

Proof. Suppose F is a maximum linear forest in T . Then $F - w_iu_i + u_ju_i$ is also a linear forest in $T[u_i \rightarrow u_j]$. Thus, we have $l(T) \leq l(T[u_i \rightarrow u_j])$. ■

Theorem 3.1. For any tree T with n vertices and diameter $2d$ ($d \geq 2$),

$$2d \leq l(T) \leq n - \left\lceil \frac{n-2}{2d-1} \right\rceil;$$

for any tree T with n vertices and diameter $2d+1$ ($d \geq 2$), if $n \geq 4d+2$ then

$$2d+1 \leq l(T) \leq n - \left\lceil \frac{n-4}{2d-1} \right\rceil;$$

if $n \leq 4d+1$ then

$$2d+1 \leq l(T) \leq n - \left\lceil \frac{n-3}{2d-1} \right\rceil;$$

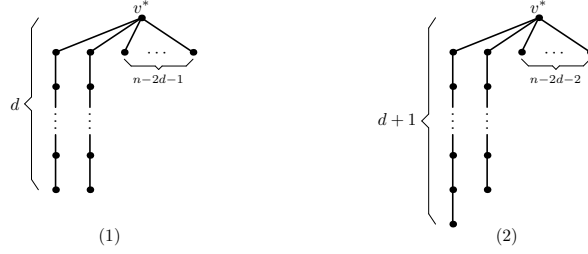


Figure 2: Extremal graphs that achieve lower bounds.

Proof. Let v^* be the center of tree T . Then T can be viewed as a rooted tree with root v^* . For the lower bounds, clearly we have $l(T) = f(L(T)) \geq t(L(T)) = d(H)$. Moreover, the extremal graph that achieve these bounds are shown in Fig.2.

For the upper bounds, we consider the extremal graphs. Partition $V(T)$ into sets $V_0(T) = \{v^*\}, V_1(T), V_2(T), \dots, V_d(T), V_{d+1}(T)$, where $V_i(T) = \{w | d(v^*, w) = i\}$. If the diameter of T is $2d$, then V_{d+1} is an empty set. In case of no confusion, we simplify $V_i(T)$ as V_i . If the diameter of T is $2d + 1$, then V_{d+1} is not empty. Let $V_{\geq 2} = V_2 \cup \dots \cup V_d \cup V_{d+1}$. We say the vertices in V_i are vertices in depth i . Let

$$\mathcal{T}_1 = \{T : |V(T)| = n, d(v) \leq 2 \text{ for any } v \in V_{\geq 2}, d(v) \leq 3 \text{ for any } v \in V_1 \text{ and } r(T) \leq d\}$$

if the diameter of T is $2d$; let

$$\mathcal{T}_1 = \{T : |V(T)| = n, d(v) \leq 2 \text{ for any } v \in V_{\geq 2}, d(v) \leq 3 \text{ for any } v \in V_1 \text{ and } r(T) \leq d + 1\}$$

if the diameter of T is $2d + 1$, where $r(T)$ is the radius of tree T . It means that \mathcal{T}_1 is a set of trees such that vertices in depth greater than one have degree less than two and vertices in depth one have degree less than three.

We claim that for any tree T there exists a tree T' in \mathcal{T}_1 such that $l(T) \leq l(T')$. Otherwise, let T be the counterexample with $|V_1|$ maximum. Clearly, T is not in \mathcal{T}_1 . Then, T has a vertex v in V_1 such that $d(v) \geq 4$ or T has a vertex v in $V_{\geq 2}$ such that $d(v) \geq 3$. We divide the proof into two cases as follows.

Case 1. T has a vertex v in V_1 such that $d(v) \geq 4$. Assume that $d(v) = t + 1$ and $t \geq 3$. Then v has one neighbor in V_0 which is root v^* and have t neighbors in V_2 . Let v_1, v_2, \dots, v_t be these t neighbors in V_2 and T_1, T_2, \dots, T_t be subtrees of v with root v_1, v_2, \dots, v_t . Suppose T has a maximum linear forest F . Then at most two edges of $v^*v, vv_1, vv_2, \dots, vv_t$ are in F . Since $t \geq 3$, there exists one of vv_1, vv_2, \dots, vv_t that is not in F . Without loss of generality, we assume vv_t is not in F . Then by removing edge vv_t from T and adding edge v^*v_t , we get a new tree \bar{T} . Clearly, we have $l(T) \leq l(\bar{T})$ since F is also a linear forest in \bar{T} . Moreover, $V_1(\bar{T})$ has more vertices than $V_1(T)$. Since T is the counterexample with $|V_1|$ maximum, we know that \bar{T} is no longer a counterexample. Therefore, there exists a tree T' in \mathcal{T}_1 such that $l(\bar{T}) \leq l(T')$. Then $l(T) \leq l(\bar{T}) \leq l(T')$ which contradicts with that T is a counterexample.

Case 2. T has a vertex v in $V_k (k \geq 2)$ such that $d(v) \geq 3$. If $d(v) \geq 4$, then we can get a contradiction by the same argument as in Case 1. Thus, we only need to consider the case $d(v) = 3$. Then v has one neighbor w in V_{k-1} and have two neighbors v_1 and v_2 in V_{k+1} . T_1, T_2 be subtrees of v with root v_1, v_2 . Suppose T has a maximum linear forest F . Then at

most two edges of wv, vv_1, vv_2 are in F . If wv is not in F , by removing edge wv from T and adding edge v^*v , we get a new tree \bar{T} . We have $l(T) \leq l(\bar{T})$ since F is also a linear forest in \bar{T} . Since $V_1(\bar{T})$ is increased by one, \bar{T} is no longer a counterexample. Therefore, there exists a tree T' in \mathcal{T}_1 such that $l(\bar{T}) \leq l(T')$. We get a contradiction. If one of vv_1 and vv_2 is not in F , without loss of generality, we assume vv_2 is not in F . Then by removing edge vv_2 from T and adding edge v^*v_2 , we get a new tree \bar{T} , which also leads to a contradiction.

Thus, we prove the claim. Let $s(T)$ be the number of degree-two vertices in $V_1(T)$ and

$$\mathcal{T}_2 = \{T : s(T) \leq 3, T \in \mathcal{T}_1\}.$$

We claim that for any tree T in \mathcal{T}_1 there exists a tree T' in \mathcal{T}_2 such that $l(T) \leq l(T')$. Otherwise, let T be the counterexample in \mathcal{T}_1 with $s(T)$ minimum. Then v^* has at least four degree-two neighbors. Assume they are v_1, v_2, \dots, v_s and paths P_1, P_2, \dots, P_s are subtrees of v^* with roots v_1, v_2, \dots, v_s , respectively. Let F be the maximum linear forest in T . Then at least two of edges $v^*v_1, v^*v_2, \dots, v^*v_s$ are not in F . Without loss of generality, we suppose v^*v_1, v^*v_2 are not in F . Then all edges in P_1, P_2 are in F . If one of paths P_1, P_2 has length at least 2. Without loss of generality, we suppose P_1 has length at least 2. Let u_1 be the other ending vertex of P_1 and w_1 be the parent of u_1 in rooted tree T . Then by removing edge w_1u_1 and adding edge v_1u_1 , we get a new tree \bar{T} . We have $l(T) \leq l(\bar{T})$ since $F - w_1u_1 + v_1u_1$ is a linear forest in \bar{T} . Since $s(\bar{T}) = s(T) - 1$, there exists a tree T' in \mathcal{T}_2 such that $l(\bar{T}) \leq l(T')$. It leads to a contradiction. If P_1, P_2 all have length one. Assume that P_1 is an edge v_1u_1 and P_2 is an edge v_2u_2 . Then by removing edge v_1u_1 and adding edge v_2u_1 , we get a new tree \bar{T} with $l(T) \leq l(\bar{T})$, which leads to a contradiction. Thus, we prove the claim.

Let

$$\mathcal{T}_3 = \{T : 2 \leq s(T) \leq 3, T \in \mathcal{T}_1\}.$$

We claim that for any tree T in \mathcal{T}_2 there exists a tree T' in \mathcal{T}_3 such that $l(T) \leq l(T')$. Suppose T is a tree in \mathcal{T}_2 and T is a counterexample with $s(T)$ maximum. Clearly, $s(T) \leq 1$. Firstly we claim T has no leaves in V_1 . Otherwise, assume v_0 is a leaf in V_1 . If there is a degree-3 vertex in V_1 . Then let u_0 is another leaf belonging to a subtree of v^* with a degree-3 vertex as root. By Leaf-exchange operation on T , we get a new tree $T[u_0 \rightarrow v_0]$ with $s(T[u_0 \rightarrow v_0]) \geq s(T) + 1$, a contradiction. If there is no degree-3 vertex in V_1 . Then V_1 has at least two leaves v_1 and v_2 since T have at least $2d$ vertices. By Leaf-exchange operation on T , we get a new tree $T[v_1 \rightarrow v_2]$ with $s(T[v_1 \rightarrow v_2]) = s(T) + 1$, a contradiction. Thus, T has no leaves in V_1 .

Let v^* be the center of tree T and we view T as a rooted tree. Let F be a maximum linear forest in T . Clearly, there is at least one degree-3 vertices in V_1 . Let v_1, v_2, \dots, v_s be degree-3 vertices in V_1 . If at least one of edges $v^*v_1, v^*v_2, \dots, v^*v_s$ is in F . Without loss of generality, suppose v^*v_1 is in F . Suppose v_1 have two neighbors u_1, u_2 in V_2 . Then at least one of edges v_1u_1 and v_1u_2 is not in F . suppose v_1u_1 is not in F . Then by removing edge v_1u_1 and adding edge v^*u_1 , we get a new tree \bar{T} with $l(T) \leq l(\bar{T})$. Clearly, $s(\bar{T}) = s(T) + 2$ and \bar{T} is in \mathcal{T}_3 . If none of $v^*v_1, v^*v_2, \dots, v^*v_s$ is in F . Then at most one adjacent edge of v^* is in F , and edges in each subtrees T_i with root v_i are all in F . Suppose v_1 have two neighbors u_1, u_2 in V_2 . Then by removing edge v_1u_1 and adding edge v^*u_1 , we get a new tree \bar{T} . Then $F - v_1u_1 + v^*u_1$ is a linear forest in \bar{T} . Clearly, $s(\bar{T}) = s(T) + 2$ and \bar{T} is in \mathcal{T}_3 .

Now we consider trees with diameter $2d$ firstly. Since any tree T in \mathcal{T}_3 has $s(T) \geq 2$.

Thus, at least two subtrees of v^* are paths with one ending vertex as root in T . We call two leaves of these two subtrees critical leaves and call all the other leaves non-critical leaves. We call a tree T with diameter d in \mathcal{T}_3 complete if and only if two critical leaves are in V_d and all but at most one non-critical leaves are in V_d . Let

$$\mathcal{T}_4 = \{T: T \text{ is complete}, T \in \mathcal{T}_3\}.$$

We claim that for any tree T in \mathcal{T}_3 there exists a tree T' in \mathcal{T}_4 such that $l(T) \leq l(T')$. Otherwise, let $T \in \mathcal{T}_3$ be a counterexample with least number of leaves that are not in V_d . Suppose T has a critical leaf which is not in V_d . Then T has a non-critical leaf since the diameter of T is $2d$. Then we can do Leaf-Exchange operation from a non-critical leaf to a critical leaf consecutively until all two critical leaves are in V_d . Thus, T only have non-critical leaves that are not in V_d . If T have two non-critical leaves w_1, w_2 that are not in V_d . Let P_1 be the longest path with one ending vertex w_1 and other ending vertex with degree two in T . Let P_2 be the longest path with one ending vertex w_2 and other ending vertex with degree two in T . We do Leaf-Exchange operation from a non-critical leaf of P_1 to a non-critical leaf of P_2 consecutively until P_1 is empty or the leaf in P_2 is in V_d . Then we get a new tree \bar{T} with less leaves that are not in V_d . Moreover, $l(T) \leq l(\bar{T})$ and \bar{T} is no longer a counterexample, a contradiction. Thus, we prove the claim.

Now, we find that trees in \mathcal{T}_4 with n vertices are uniquely determined, which has been shown in Fig.3. The unique tree in \mathcal{T}_4 with n vertices and diameter $2d$ is the extremal graph that achieve the upper bounds, we call it T^* . If the remainder is between 1 and d when $n - 2d - 1$ is divided by $2d - 1$, then $s(T^*) = 3$ as shown in Fig.3(1). If the remainder is between $d + 1$ and $2d$ when $n - 2d - 1$ is divided by $2d - 1$, then $s(T^*) = 2$ as shown in Fig.3(2).

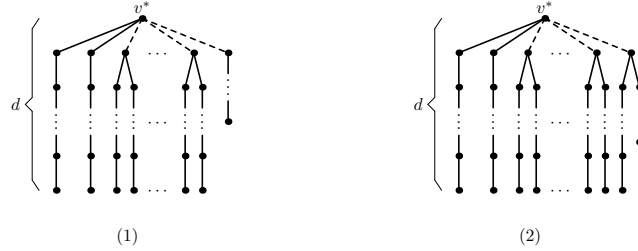


Figure 3: Extremal graphs that achieve upper bounds.

For the first case, we have $ex(v^*) = 1$ and $out(T^*) = 2 \left\lfloor \frac{n-2d-1}{2d-1} \right\rfloor + 3 = 2 \left\lfloor \frac{n-2}{2d-1} \right\rfloor + 1$. By Lemma 3.1, we have

$$\begin{aligned}
l(T^*) &\leq n - h(T^*) \\
&\leq n - \left\lceil \frac{out(T^*) + \sum_{v \in V_{in}} ex(v)}{2} \right\rceil \\
&= n - \left\lfloor \frac{n-2}{2d-1} \right\rfloor - 1 \\
&= n - \left\lceil \frac{n-2}{2d-1} \right\rceil.
\end{aligned}$$

However, by removing the dashed edges as shown in Fig.3(1), we get a linear forest with $n - 1 - \left\lceil \frac{n-2d-1}{2d-1} \right\rceil = n - \left\lceil \frac{n-2}{2d-1} \right\rceil$ edges. Thus, $l(T^*) = n - \left\lceil \frac{n-2}{2d-1} \right\rceil$.

For the second case, we have $ex(v^*) = 0$ and $out(T^*) = 2 \left\lceil \frac{n-2d-1}{2d-1} \right\rceil + 2 = 2 \left\lceil \frac{n-2}{2d-1} \right\rceil$. By Lemma 3.1, we have

$$\begin{aligned} l(T^*) &\leq n - h(T^*) \\ &\leq n - \left\lceil \frac{out(T^*) + \sum_{v \in V_{in}} ex(v)}{2} \right\rceil \\ &= n - \left\lceil \frac{n-2}{2d-1} \right\rceil. \end{aligned}$$

However, by removing the dashed edge as shown in Fig.3(2), we get a linear forest with $n - 1 - \left\lceil \frac{n-2d-1}{2d-1} \right\rceil = n - \left\lceil \frac{n-2}{2d-1} \right\rceil$ edges. Thus, $l(T^*) = n - \left\lceil \frac{n-2}{2d-1} \right\rceil$.

Combining above two cases, we prove that for any tree T with n vertices and diameter $2d$ ($d \geq 2$),

$$l(T) \leq l(T^*) \leq n - \left\lceil \frac{n-2}{2d-1} \right\rceil.$$

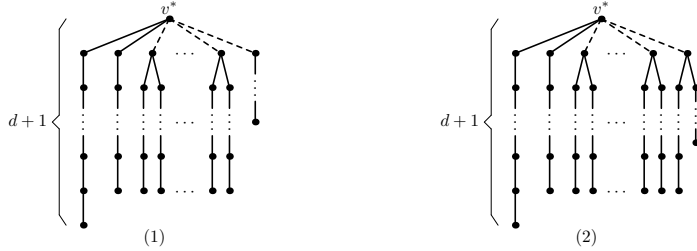


Figure 4: Extremal graphs that achieve upper bounds.

For trees with diameter $2d + 1$, let T be a tree in \mathcal{T}_3 with diameter $2d + 1$. Then vertices in V_{d+1} are all leaves. Moreover, these leaves are all in the same subtree of v^* otherwise the diameter of T will become to $2d + 2$. Thus, we have $|V_{d+1}| = 1$ or $|V_{d+1}| = 2$.

If $|V_{d+1}| = 1$, we denote this vertex as u and its neighbor as w . By remove this vertex, we get a tree T' with diameter $2d$. Suppose F is a maximum linear forest in T . Then $F - uw$ is a linear forest in T' . Suppose F' is a maximum linear forest in T' . Then $F' + uw$ is a linear forest in T . Then we have $l(T) = l(T') + 1$. Thus, we can a graph T_1^* with n vertices and diameter $2d + 1$ as shown in Fig.4. And any tree T in \mathcal{T}_3 with $|V_{d+1}| = 1$ has $l(T) \leq l(T_1^*)$. In this case, we have

$$\begin{aligned} l(T_1^*) &= (n-1) - \left\lceil \frac{n-1-2}{2d-1} \right\rceil + 1 \\ &= n - \left\lceil \frac{n-3}{2d-1} \right\rceil. \end{aligned}$$

If $|V_{d+1}| = 2$, we denote these two vertex as u_1 and u_2 and their neighbors as w_1 and w_2 , respectively. Clearly, $n \geq 4d + 2$. By remove these two vertex, we get a tree T' with diameter $2d$. Similarly, we have $l(T) = l(T') + 2$. Thus, we can get a graph T_2^* with n vertices and diameter $2d + 1$ as shown in Fig.5. And any tree T in \mathcal{T}_3 with $|V_{d+1}| = 2$ has $l(T) \leq l(T_2^*)$. In this case, we have

$$\begin{aligned} l(T_2^*) &= (n - 2) - \left\lceil \frac{n - 2 - 2}{2d - 1} \right\rceil + 2 \\ &= n - \left\lceil \frac{n - 4}{2d - 1} \right\rceil. \end{aligned}$$

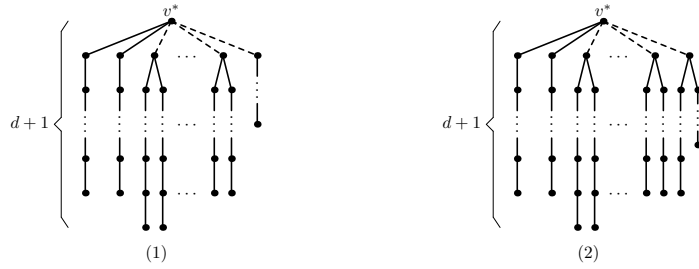


Figure 5: Extremal graphs that achieve upper bounds.

Since $l(T_1^*) \leq l(T_2^*)$. Then we have $l(T) \leq l(T_2^*) = n - \left\lceil \frac{n-4}{2d-1} \right\rceil$ if $n \geq 4d + 2$ and T_2^* is the extremal graph that achieve the upper bound. $l(T) \leq l(T_1^*) = n - \left\lceil \frac{n-3}{2d-1} \right\rceil$ if $n \leq 4d + 1$ and T_1^* is the extremal graph that achieve the upper bound. ■

4 The algorithm aspects

Let v^* be the center of tree T . Then T can be viewed as a rooted tree with root v^* . Let T_1, T_2, \dots, T_k be subtrees of v^* with root v_1, v_2, \dots, v_k , respectively. Let $F(T)$ be the edge set of a maximum linear forest in T . Let $F'(T)$ be the edge set of a maximum linear forest of T such that the degree of v^* is at most one.

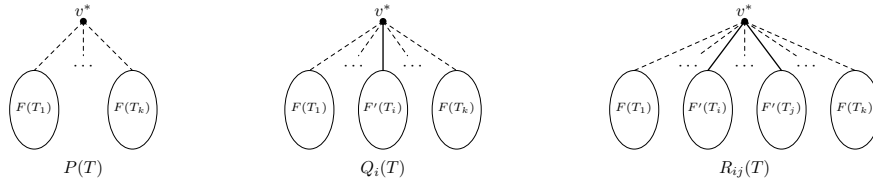


Figure 6: The structures of linear forests $P(T)$, $Q_i(T)$ and $R_{ij}(T)$.

Then, we define three kinds of linear forests $P(T)$, $Q_i(T)$ and $R_{ij}(T)$ as follows.

$$\begin{cases} P(T) = \cup_{i=1}^k F(T_i), \\ Q_i(T) = (\cup_{r \neq i} F(T_r)) \cup F'(T_i) \cup \{(v^*, v_i)\}, \\ R_{ij}(T) = (\cup_{r \neq i, j} F(T_r)) \cup F'(T_i) \cup F'(T_j) \cup \{(v^*, v_i), (v^*, v_j)\}. \end{cases}$$

As shown in Fig.6, $P(T)$ is a linear forest in T such that v^* has degree zero; $Q_i(T)$ is a linear forest such that v^* has degree one and (v^*, v_i) is an edge in the linear forest; $R_{ij}(T)$ is a linear forest such that v^* has degree two and $(v^*, v_i), (v^*, v_j)$ are edges in the linear forest. Let $S(T)$ be the largest linear forest among all $P(T)$, $Q_i(T)$ and $R_{ij}(T)$. Let $S'(T)$ be the largest linear forest among all $P(T)$, $Q_i(T)$.

Theorem 4.1. *For any tree T , $S(T)$ is a maximum linear forest in T , $S'(T)$ is a maximum linear forest in T such that root v^* has degree at most one.*

Proof. For any tree T , if $S(T)$ is not a maximum linear forest in T . Then suppose $F(T)$ is a maximum linear forest in T . We can divide the proof into three cases according the degree of v^* in $F(T)$.

Case 1. If the degree of v^* in $F(T)$ is zero. Then let $F[T_r] = F(T) \cap E(T_r)$, for $r = 1, 2, \dots, k$. Clearly, $F[T_r]$ is a linear forest in subtree T_r . Thus, $|F[T_r]| \leq |F(T_r)|$. Therefore, $|F(T)| \leq |P(T)|$.

Case 2. If the degree of v^* in $F(T)$ is one. Suppose v^*v_i is in $F(T)$. Then let $F[T_r] = F(T) \cap E(T_r)$, for $r = 1, 2, \dots, k$. Clearly, for $r \neq i$, $F[T_r]$ is a linear forest in subtree T_r and $F[T_i]$ is a linear forest in subtree T_i with degree of v^* at most one. Thus, for each $r \neq i$ we have $|F[T_r]| \leq |F(T_r)|$ and $|F[T_i]| \leq |F'(T_i)|$. Therefore, $|F(T)| \leq |Q_i(T)|$.

Case 3. If the degree of v^* in $F(T)$ is two. Suppose v^*v_i and v^*v_j are in $F(T)$. Then let $F[T_r] = F(T) \cap E(T_r)$. Clearly, for $r \neq i, j$, $F[T_r]$ is a linear forest in subtree T_r . $F[T_i]$ is a linear forest in subtree T_i with degree of v^* at most one and $F[T_j]$ is a linear forest in subtree T_j with degree of v^* at most one. Thus, for each $r \neq i, j$ we have $|F[T_r]| \leq |F(T_r)|$. For each $r = i, j$, we have $|F[T_i]| \leq |F'(T_i)|$ and $|F[T_j]| \leq |F'(T_j)|$. Therefore, $|F(T)| \leq |R_{ij}(T)|$.

Combining these cases, we get the conclusion that $F(T) \leq S(T)$, which implies that $S(T)$ is a maximum linear forest in T . Similarly, we can prove $S'(T)$ is a maximum linear forest in T such that root v^* has degree at most one. ■

According this recursive structure of tree T , it is easy to design a dynamic programming algorithm for a rooted tree from bottom to top. Therefore, we get a polynomial time algorithm for finding maximum linear forest in trees and finding minimum feedback vertex set in line graph of trees.

5 Line graphs of full k -ary trees

In graph theory, a *full k -ary tree* is a rooted tree where within each level every node has either 0 or k children. A *complete k -ary tree* is a full k -ary tree which is completely filled on every level except for the last level. A *perfect k -ary tree* is a full k -ary tree in which all leaf nodes are at the same depth. Moreover, We call a full k -ary tree *linear k -ary tree* if it has only one internal vertex on every level.

Lemma 5.1. *For any full k -ary tree T with n vertices,*

$$\frac{n+k-1}{k} \leq l(T) \leq \frac{2n-2}{k}.$$

Proof. Suppose T has x internal vertices and y leaves. Then we have $kx + 1 = x + y = n$. It follows that $x = \frac{n-1}{k}$ and $y = \frac{(k-1)n+1}{k}$. By Lemma 3.1, we have $h(T) \leq \frac{(k-1)n+1}{k} - 1$. Thus, $l(T) \geq n - h(T) = \frac{n+k-1}{k}$.

For the upper bound, we can divide $n - 1$ edges of T into $\frac{n-1}{k}$ groups such that each k edges with the same parent are in the same group. Since the degree of vertices in the linear forest is at most two. Thus, at most two edges in each group are in the linear forest. Therefore, we get $l(T) \leq \frac{2n-2}{k}$. \blacksquare

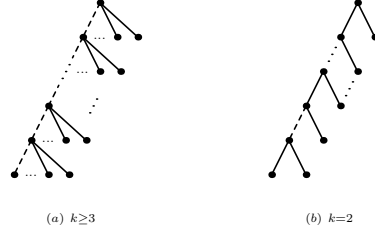


Figure 7: Linear k -ary trees and their maximum linear forests.

Let T be a linear k -ary tree with n vertices. Suppose v_i be the internal vertex at depth i and root v_0 is at depth 0. If $k \geq 3$, then each v_i have two adjacent edges that link to leaves. Clearly, as shown in Fig.7(a) we have $l(T) = \frac{2n-2}{k}$. If $k \geq 2$, by Theorem 4.1 it is easy to check that the linear forest shown in Fig.7(b) is maximum. Thus, $l(T) = \frac{3(n-1)}{4}$ if $\frac{n-3}{2}$ is odd; $l(T) = \frac{3n-1}{4}$ if $\frac{n-3}{2}$ is even.

Theorem 5.1. For any perfect k -ary tree T with n vertices, $l(T) = \frac{2n-1+(-1)^h}{k+1}$, where $h = \log_k(n(k-1)+1)$ is the height of T and leaves are at height 1.

Proof. Let $h = \log_k(n(k-1)+1)$ be the height of T . We construct a linear forest of T as follows. Firstly, we choose two vertex-disjoint paths of length h that go from root to two leaves. Then tree T is decomposed into $k-2$ subtrees of height $h-1$, $2(k-1)$ subtrees of height $h-2$, $2(k-1)$ subtrees of height $h-3$, \dots , $2(k-1)$ subtrees of height 3 and $2(k-1)$ subtrees of height 2 as shown in Fig.8. We can get a linear forest of T by choosing two vertex-disjoint paths that go from root to two leaves in each subtree recursively. Denote the obtained linear forest in perfect perfect k -ary tree of height h by F_h . Let f_h be the number of edges in F_h . Then it is easy to see that $f_1 = 0$ and $f_2 = 2$. According the recursive construction of the obtained linear forest, we have

$$f_h = (k-2)f_{h-1} + 2(k-1) \sum_{i=1}^{h-2} f_i + 2(h-1),$$

and

$$f_{h-1} = (k-2)f_{h-2} + 2(k-1) \sum_{i=1}^{h-3} f_i + 2(h-2).$$

Combine two equations, we get a recursive relation

$$f_h = (k-1)f_{h-1} + kf_{h-2} + 2.$$

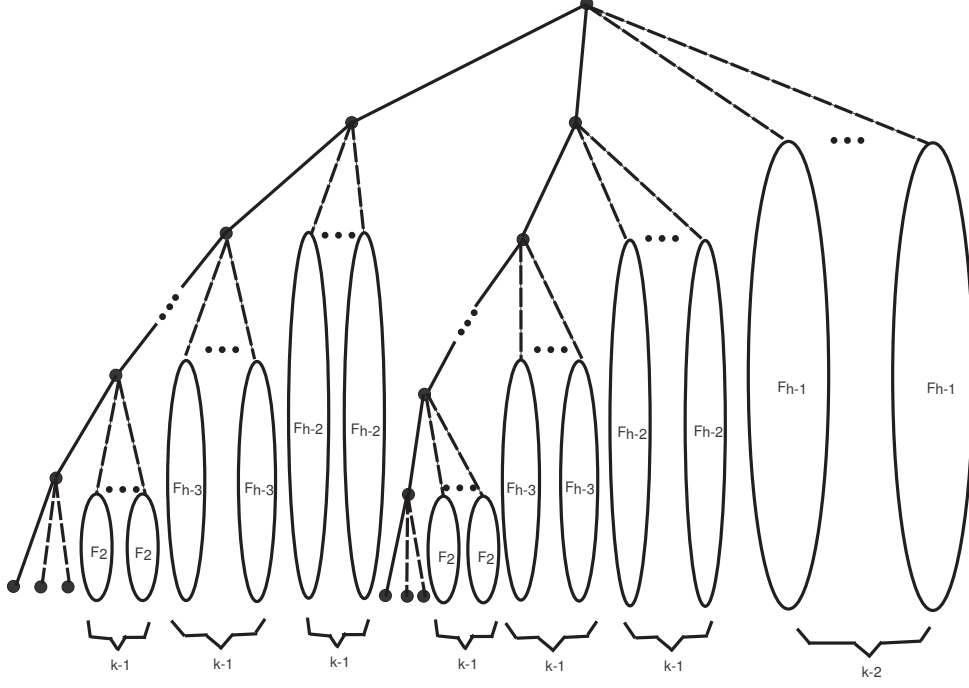


Figure 8: A linear forest in perfect k -ary trees.

Let $F(x)$ be the generating function satisfy $F(x) = \sum_{h \geq 2} f_h x^h$. Then,

$$\begin{aligned}
 F(x) &= \sum_{h \geq 2} f_h x^h \\
 &= 2x^2 + \sum_{h \geq 3} f_h x^h \\
 &= 2x^2 + \sum_{h \geq 3} ((k-1)f_{h-1} + kf_{h-2} + 2)f_h x^h \\
 &= 2 \sum_{h \geq 2} x^h + (k-1) \sum_{h \geq 3} f_{h-1} x^h + k \sum_{h \geq 3} f_{h-2} x^h \\
 &= \frac{2x^2}{1-x} + (k-1)x F(x) + kx^2 F(x)
 \end{aligned}$$

Therefore, we have

$$F(x) = \frac{2x^2}{(1-x)(1+x)(1-kx)}.$$

Suppose that

$$F(x) = \frac{A}{1-x} + \frac{B}{1+x} + \frac{C}{1-kx}.$$

Determine coefficients A , B and C , we have

$$A = \frac{1}{1-k}, B = \frac{1}{k+1}, C = \frac{2}{k^2-1}.$$

Since

$$n = 1 + k + k^2 + \dots + k^{h-1} = \frac{k^h - 1}{k - 1}.$$

It follows that $k^m = n(k - 1) + 1$. Therefore,

$$\begin{aligned} f_h &= \frac{1 + k}{1 - k^2} + \frac{(-1)^h}{k + 1} + \frac{2k^h}{k^2 - 1} \\ &= \frac{2(n(k - 1) + 1) - k - 1}{k^2 - 1} + \frac{(-1)^h}{k + 1} \\ &= \frac{2n - 1 + (-1)^h}{k + 1}. \end{aligned}$$

We get a linear forest with $\frac{2n-1+(-1)^h}{k+1}$ edges in perfect k -ary tree T with n vertices and height h . Thus, $l(T) \geq f_h = \frac{2n-1+(-1)^h}{k+1}$.

Now we need to prove that F_h is the maximum linear forest in perfect k -ary tree T with height h . Let v^* be the root of T and T_1, T_2, \dots, T_k be the k subtrees of v^* with root v_1, v_2, \dots, v_k . Since T is a perfect k -ary tree, each subtree T_i is identical to a perfect k -ary tree with n vertices height $h - 1$. Suppose v^*v_1 and v^*v_2 are in linear forest F_h . Let F'_{h-1} be the subset of F_h in subtree T_1 and let f'_{h-1} be the number of edges in F'_{h-1} . Then F'_{h-1} is a linear forest in T_1 such that $d(v_1) = 1$.

We claim that in perfect k -ary tree with height h , F'_h is the maximum linear forest such that degree of the root is at most one and F_h is the maximum linear forest. We prove the claim by induction on h . For $h = 1$ and $h = 2$, it is easy to check F'_1, F'_2 are maximum linear forests with degree of root at most one and F_1, F_2 are maximum linear forests, where F'_1, F_1 are empty sets. Suppose the claim is true for perfect k -ary tree with height $h - 1$. Let T be a perfect k -ary tree with height h . Then, by Theorem 4.1 we have

$$\begin{cases} P(T) = \cup_{i=1}^k F(T_i), \\ Q_i(T) = (\cup_{r \neq i} F(T_r)) \cup F'(T_i) \cup \{(v^*, v_i)\}, \\ R_{ij}(T) = (\cup_{r \neq i, j} F(T_r)) \cup F'(T_i) \cup F'(T_j) \cup \{(v^*, v_i), (v^*, v_j)\}. \end{cases}$$

By induction hypothesis each $F(T_i)$ is identical to F_{h-1} and each $F'(T_i)$ is identical to F'_{h-1} . Thus, $Q_i(T)$ is identical to F'_h and $R_{ij}(T)$ is identical to F_h . Suppose T has n vertices. Then each T_i has $\frac{n-1}{k}$ vertices. Moreover, F_h are consist of $k - 2$ F_{h-1} 's, two F'_{h-1} 's and two extra edges. Therefore, we have $f'_{h-1} = \frac{1}{2}(f_h - (k - 2)f_{h-1} - 2)$. Then

$$\begin{cases} |P(T)| = kf_{h-1}, \\ |Q_i(T)| = (k - 1)f_{h-1} + f'_{h-1} + 1, \\ |R_{ij}(T)| = (k - 2)f_{h-1} + 2f'_{h-1} + 2 = f_h. \end{cases}$$

Since

$$f'_{h-1} + 1 - f_{h-1} = \frac{1}{2}(f_h - (k - 2)f_{h-1} - 2) + 1 - f_{h-1}$$

$$\begin{aligned}
&= \frac{1}{2}(f_h - kf_{h-1}) \\
&= \frac{1}{2} \left(\frac{2n-1+(-1)^h}{k+1} - k \frac{2^{\frac{n-1}{k}} - 1 + (-1)^{h-1}}{k+1} \right) \\
&= 1 - (-1)^{h-1} \geq 0.
\end{aligned}$$

It follows that $|P(T)| \leq |Q_i(T)| \leq |R_{ij}(T)|$. By Theorem 4.1, we see $R_{ij}(T)$ is the maximum linear forest in T and $Q_i(T)$ is maximum linear forest such that degree of the root is at most one. We prove the claim and $l(T) = \frac{2n-1+(-1)^h}{k+1}$. ■

Finally, we end up this paper by proposing a conjecture on maximum linear forests in full k -ary trees.

Conjecture 5.1. *For any full k -ary tree T on n vertices, $l(T_1) \leq l(T) \leq l(T_2)$, in which T_1 is the complete k -ary tree on n vertices and T_2 is the linear k -ary tree on n vertices.*

References

- [1] P. Erdős, M. Saks and V. T. Sós, Maximum induced trees in graphs, *J. Combin. Theory Ser. B* 41 (1986), 61–79.
- [2] J. Fox, P. S. Loh and B. Sudakov, Large induced trees in K_r -free graphs. *J. Combin. Theory Ser. B*, 99(2) (2009), 494–501.
- [3] V. Bafna, P. Berman and T. Fujito, A 2-approximation algorithm for the undirected feedback vertex set problem, *SIAM J. Discrete Math.* 12 (1999), 289–297.
- [4] L. W. Beineke and R. C. Vandell, Decycling graphs, *J. Graph Theory* 25 (1997), 59–77.
- [5] M. R. Garey and D. S. Johnson, Computers and Intractability, *A guide to the theory of NP-completeness. A Series of Books in the Mathematical Sciences. W. H. Freeman and Co.*, 1979. x+338 pp. ISBN: 0-7167-1045-5.
- [6] R. M. Karp, Reducibility among combinatorial problems, *Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, 1972)*, pp. 85–103. Plenum, 1972.
- [7] A. Shamir, A linear time algorithm for finding minimum cutsets in reducible graphs. *SIAM Journal on Computing*, 8 (1979), 645–655.
- [8] C.-C. Wang, E. L. Lloyd and M. L. Soffa, Feedback vertex sets and cyclically reducible graphs, *J. Assoc. Comput. Mach.* 32 (1985), 296–313.
- [9] Y.D. Liang, On the feedback vertex set in permutation graphs. *Inform. Process. Lett.* 52 (1994), 123–129.
- [10] C. L. Lu, C. Y. Tang, A linear-time algorithm for the weighted feedback vertex problem on interval graphs. *Inform. Process. Lett.* 61 (1997), 107–111.

- [11] S. R. Coorg, C. P. Rangan, Feedback vertex set on cocomparability graphs. *Networks*, 26(2) (1995), 101–111.
- [12] Y. D. Liang , M. S. Chang, Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs. *Acta Informatica*, 34 (1997), 337–346.
- [13] D. Kratsch, H. Ller, I. Todinca, Feedback vertex set on AT-free graphs. *Discr. Appl. Math*, 156 (2008), 1936–1947.